

SS POSTTEST 6

```

1  import java.util.ArrayList;
2  import java.util.InputMismatchException;
3  import java.util.Scanner;
4
5  interface Peminjaman {
6      void pinjam();
7      void kembalikan();
8  }
9
10 abstract class Buku {
11     protected String judul;
12     protected String penulis;
13     protected boolean tersedia;
14     protected final String kodeUnik = "MB123";
15     protected static int totalBuku = 0; // static variable
16
17     public Buku(String judul, String penulis) {
18         this.judul = judul;
19         this.penulis = penulis;
20         this.tersedia = true;
21         totalBuku++;
22     }
23
24     public abstract String getInfo();
25
26     public void setJudul(String judul) {
27         this.judul = judul;
28     }
29
30     public void setPenulis(String penulis) {
31         this.penulis = penulis;
32     }
33
34     public static int getTotalBuku() {
35         return totalBuku;
36     }
37 }
38
39 // Class BukuFisik mengimplementasi interface
40 class BukuFisik extends Buku implements Peminjaman {
41     private String lokasiRak;
42
43     public BukuFisik(String judul, String penulis, String lokasiRak) {
44         super(judul, penulis);
45         this.lokasiRak = lokasiRak;
46     }
47
48     @Override
49     public String getInfo() {
50         return judul + " - " + penulis + " [Fisik | Rak: " + lokasiRak + "] " + (tersedia ? "(Tersedia)" : "(Dipinjam)");
51     }
52

```

```

53     @Override
54     public void pinjam() {
55         if (tersedia) {
56             tersedia = false;
57             System.out.println("Buku fisik berhasil dipinjam.");
58         } else {
59             System.out.println("Buku sedang tidak tersedia.");
60         }
61     }
62
63     @Override
64     public void kembalikan() {
65         if (!tersedia) {
66             tersedia = true;
67             System.out.println("Buku fisik berhasil dikembalikan.");
68         } else {
69             System.out.println("Buku sudah tersedia.");
70         }
71     }
72 }
73
74 // Final class BukuDigital juga mengimplementasi interface
75 final class BukuDigital extends Buku implements Peminjaman {
76     private double ukuranFileMB;
77
78     public BukuDigital(String judul, String penulis, double ukuranFileMB) {
79         super(judul, penulis);
80         this.ukuranFileMB = ukuranFileMB;
81     }
82
83     @Override
84     public String getInfo() {
85         return judul + " - " + penulis + " [Digital | " + ukuranFileMB + " MB] " + (tersedia ? "(Tersedia)" : "(Dipinjam)");
86     }
87
88     @Override
89     public void pinjam() {
90         if (tersedia) {
91             tersedia = false;
92             System.out.println("Buku digital berhasil dipinjam.");
93         } else {
94             System.out.println("Buku sedang tidak tersedia.");
95         }
96     }
97
98     @Override
99     public void kembalikan() {
100         if (!tersedia) {
101             tersedia = true;
102             System.out.println("Buku digital berhasil dikembalikan.");
103         } else {
104             System.out.println("Buku sudah tersedia.");
105         }
106     }
107 }
108
109 public class MariBaca {
110     private static ArrayList<Buku> daftarBuku = new ArrayList<>();
111     private static Scanner scanner = new Scanner(System.in);
112

```

```

113     public static void main(String[] args) {
114         while (true) {
115             try {
116                 System.out.println("\n=== Layanan Pendataan Buku 'MariBaca' ===");
117                 System.out.println("1. Tambah Buku");
118                 System.out.println("2. Tampilkan Buku");
119                 System.out.println("3. Perbarui Buku");
120                 System.out.println("4. Hapus Buku");
121                 System.out.println("5. Pinjam Buku");
122                 System.out.println("6. Kembalikan Buku");
123                 System.out.println("7. Lihat Total Buku");
124                 System.out.println("8. Keluar");
125                 System.out.print("Pilih menu: ");
126                 int pilihan = scanner.nextInt();
127                 scanner.nextLine();
129                 switch (pilihan) {
130                     case 1: tambahBuku(); break;
131                     case 2: tampilkanBuku(); break;
132                     case 3: perbaruiBuku(); break;
133                     case 4: hapusBuku(); break;
134                     case 5: pinjamBuku(); break;
135                     case 6: kembalikanBuku(); break;
136                     case 7: System.out.println("Total buku: " + Buku.getTotalBuku()); break;
137                     case 8:
138                         System.out.println("Terima kasih telah menggunakan layanan MariBaca!");
139                         return;
140                     default: System.out.println("Pilihan tidak valid.");
141                 }
142             } catch (InputMismatchException e) {
143                 System.out.println("Input tidak valid. Harus berupa angka.");
144                 scanner.nextLine(); // buang input error
145             }
146         }
147     }

149     private static final void tampilkanBuku() {
150         if (daftarBuku.isEmpty()) {
151             System.out.println("Belum ada buku.");
152             return;
153         }
154         System.out.println("\nDaftar Buku:");
155         for (int i = 0; i < daftarBuku.size(); i++) {
156             System.out.println((i + 1) + ". " + daftarBuku.get(i).getInfo());
157         }
158     }

159
160     private static void tambahBuku(String judul, String penulis, String lokasiRak) {
161         daftarBuku.add(new BukuFisik(judul, penulis, lokasiRak));
162         System.out.println("Buku fisik ditambahkan.");
163     }
164
165     private static void tambahBuku(String judul, String penulis, double ukuranFileMB) {
166         daftarBuku.add(new BukuDigital(judul, penulis, ukuranFileMB));
167         System.out.println("Buku digital ditambahkan.");
168     }

```

```
170     private static void tambahBuku() {
171         try {
172             System.out.println("Jenis buku: 1. Fisik 2. Digital");
173             System.out.print("Pilihan: ");
174             int jenis = scanner.nextInt();
175             scanner.nextLine();
176
177             System.out.print("Judul: ");
178             String judul = scanner.nextLine();
179             System.out.print("Penulis: ");
180             String penulis = scanner.nextLine();
181
182             if (jenis == 1) {
183                 System.out.print("Lokasi rak: ");
184                 String lokasi = scanner.nextLine();
185                 tambahBuku(judul, penulis, lokasi);
186             } else if (jenis == 2) {
187                 System.out.print("Ukuran file (MB): ");
188                 double ukuran = scanner.nextDouble();
189                 scanner.nextLine();
190                 tambahBuku(judul, penulis, ukuran);
191             } else {
192                 System.out.println("Jenis tidak valid.");
193             }
194         }
195     }
196 }
```



```

194         } catch (InputMismatchException e) {
195             System.out.println("Input salah. Silakan coba lagi.");
196             scanner.nextLine();
197         }
198     }
199
200     private static void perbaruiBuku() {
201         tampilkanBuku();
202         if (daftarBuku.isEmpty()) return;
203
204         System.out.print("Nomor buku yang ingin diperbarui: ");
205         int index = scanner.nextInt() - 1;
206         scanner.nextLine();
207
208         if (index < 0 || index >= daftarBuku.size()) {
209             System.out.println("Nomor tidak valid.");
210             return;
211         }
212
213         Buku buku = daftarBuku.get(index);
214         System.out.print("Judul baru: ");
215         String judul = scanner.nextLine();
216         System.out.print("Penulis baru: ");
217         String penulis = scanner.nextLine();
218
219         buku.setJudul(judul);
220         buku.setPenulis(penulis);
221
222         System.out.println("Buku diperbarui!");
223     }
224
225     private static void hapusBuku() {
226         tampilkanBuku();
227         if (daftarBuku.isEmpty()) return;
228
229         System.out.print("Nomor buku yang ingin dihapus: ");
230         int index = scanner.nextInt() - 1;
231         scanner.nextLine();
232
233         if (index < 0 || index >= daftarBuku.size()) {
234             System.out.println("Nomor tidak valid.");
235             return;
236         }

```

```

237
238     daftarBuku.remove(index);
239     System.out.println("Buku berhasil dihapus!");
240 }
241
242 private static void pinjamBuku() {
243     tampilkanBuku();
244     if (daftarBuku.isEmpty()) return;
245
246     System.out.print("Nomor buku yang ingin dipinjam: ");
247     int index = scanner.nextInt() - 1;
248     scanner.nextLine();
249
250     if (index < 0 || index >= daftarBuku.size()) {
251         System.out.println("Nomor tidak valid.");
252         return;
253     }
254
255     if (daftarBuku.get(index) instanceof Peminjaman) {
256         ((Peminjaman) daftarBuku.get(index)).pinjam();
257     }
258 }
259
260 private static void kembalikanBuku() {
261     tampilkanBuku();
262     if (daftarBuku.isEmpty()) return;
263
264     System.out.print("Nomor buku yang ingin dikembalikan: ");
265     int index = scanner.nextInt() - 1;
266     scanner.nextLine();
267
268     if (index < 0 || index >= daftarBuku.size()) {
269         System.out.println("Nomor tidak valid.");
270         return;
271     }
272
273     if (daftarBuku.get(index) instanceof Peminjaman) {
274         ((Peminjaman) daftarBuku.get(index)).kembalikan();
275     }
276 }
277 }

```